Interview Summary	Application No.	Applicant(s)	
	10/015,959	EGEVANG, KJELD B.	
	Examiner	Art Unit	
	CHUONG T. HO	2616	

All participants (applicant, applicant's representative, PTO personnel):

(1) *CHUONG T. HO*. (3)_____.

(2) *Mr. Robert V. Racunas (724) 933-9344*. (4)_____.

Date of Interview: *05 July 2007*.

Type: a)☒ Telephonic b)☐ Video Conference
c)☐ Personal [copy given to: 1)☐ applicant 2)☐ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes e)☐ No.
If Yes, brief description: _____.

Claim(s) discussed: _____.

Identification of prior art discussed: _____.

Agreement with respect to the claims f)☐ was reached. g)☐ was not reached. h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *the applicant agreed to replace the claimed amendment filed 06/20/07 by the claimed amendment filed 06/11/07 (attached the claim amendmend filed 06/11/07 for issuing)*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.

_____
Examiner's signature, if required

KACVINSKY LLC
INTELLECTUAL PROPERTY COUNSELORS

FAX COVER SHEET

DATE: June 12, 2007

TO: Examiner Chuong T. Ho

FROM: Robert Racunas (724-933-9344)

FAX NUMBER: 571-273-3133

PAGES (Including Cover Sheet): 9

COMMENTS:

4500 BROOKTREE ROAD, SUITE 102 | WEXFORD, PENNSYLVANIA 15090
PHONE: 724.933.5529 | FAX: 724.933.0028 | www.kacvinskylaw.com

PATENT P11637

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	10/015,959	Confirmation No.:	6990
Applicant	:	Egevang		
Filed	:	November 1, 2001		
TC/A.U.	:	2664		
Examiner	:	Ho, Chuong T.		
Docket No.	:	42390.P11637		

PROPOSED CLAIM AMENDMENTS

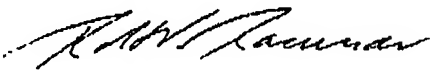
Applicant submits the following proposed claim amendments to be discussed in a telephone interview with Patent Examiner Chuong T. Ho and Supervisory Patent Examiner Huy D. Vu.

Applicant submits that the proposed claim amendments would be entitled to entry after final at least because: (i) the proposed claims amendments clearly would place the case in condition for allowance and/or (ii) the proposed claim amendments would place the case in better form for appeal.

Applicant respectfully directs the Examiner to MPEP § 714.13 and reminds the Examiner that the refusal to enter the proposed amendment should not be arbitrary. The proposed amendment should be given sufficient consideration to determine whether the claims are in condition for allowance and/or whether the issues on appeal are simplified.

Respectfully submitted,

KACVINSKY LLC



Robert V. Racunas, Reg. No. 43,027
Under 37 CFR 1.34(a)

Dated: June 11, 2007
4500 Brooktree Road, Suite 102
Wexford, PA 15090

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A method to manage packet fragmentation for address translation, comprising:

receiving a plurality of packet fragments for a packet having a first address, wherein each packet fragment includes a packet fragment header having a packet offset value and a more bit, said offset value representing a position from a starting position to an ending position of said packet, said more bit set to a predetermined position indicating that more packet fragments are to follow;

translating said first address into a second address without reassembling said packet fragments into said packet;

determining whether all packet fragments for said packet have been received by determining a status of said more bit for each packet fragment, collecting offset values in a verification table, indexing collected offset values by position in said verification table, and evaluating said collected offset values to identify any missing positions between said starting position and said ending position;

sending a verification message once all entries in said verification table are filled;

translating said first address into a second address, without reassembling said packet fragments into said packet, in response to said verification message; and

sending said packet fragments using said second address.

2. (Previously Presented) The method of claim 1, wherein said translating comprises:

identifying said packet fragment having a packet header, with said packet header having a packet identifier, translation information and a packet length;

retrieving translation information from said packet header; and

translating said first address into said second address using said translation information.

3. (Original) The method of claim 2, wherein said translation information comprises a port number.

4. (Previously Presented) The method of claim 2, wherein each of said packet fragment includes said packet fragment header having said packet identifier, and said determining comprises:

storing each packet fragment having said packet identifier and said more bits set to predetermined values.

5. (Previously Presented) The method of claim 1, wherein each offset value represents a position for said packet fragment in said packet, and said determining whether all packet fragments for said packet have been received using said offset values comprises:

retrieving said packet length; and

determining whether all positions for said packet are filled by said collected offset values using said packet length.

6. (Previously Presented) The method of claim 2, wherein each packet fragment includes a packet fragment header having said packet identifier and said determining comprises:

storing each packet fragment having said packet identifier and said offset value is a value other than zero; and

determining whether all packet fragments for said packet have been received using said offset values.

7. (Previously Presented) The method of claim 6, wherein each offset value represents a position for said packet fragment in said packet, and said determining

whether all packet fragments for said packet have been received using said offset values comprises:

retrieving said packet length; and

determining whether all positions are filled by said collected offset values using said packet length.

8. (Original) The method of claim 5, wherein each offset value represents a position in bytes divided by eight for said packet fragment in said packet.

9. (Original) The method of claim 7, wherein each offset value represents a position in bytes divided by eight for said packet fragment in said packet.

10. (Original) The method of claim 1, further comprising: detecting an occurrence of a terminating condition prior to receiving all of said packet fragments for said packet; and releasing said packet fragments in accordance with said detection.

11. (Currently Amended) A packet fragmentation manager to manage packet fragmentation for address translation, comprising:

a collection module to collect and store a plurality of packet fragments for a packet having a first address, wherein each packet fragment includes a packet fragment header having a packet offset value and a more bit, said offset value representing a position from a starting position to an ending position of said packet, said more bit set to a predetermined position indicating that more packet fragments are to follow;

a verification module to verify all packet fragments for said packet have been received; and

a translation module to retrieve translation information from one of said packet fragments and to translate said first address into a second address using said translation information, without reassembling said packet fragments into said packet,

wherein said verification module is to determine whether all packet fragments for said packet have been received by determining a status of said more bit for each packet fragment, collecting offset values in a verification table, indexing collected offset values

by position in said verification table, and evaluating said collected offset values to identify any missing positions between said starting position and said ending position;

wherein said verifications module is to send a verification message to said translation module once all entries in said verification table are filled, and said translation module is to translate said first address into said second address, without reassembling said packet fragments into said packet, in response to said verification message.

12. (Previously Presented) The packet fragmentation manager of claim 11, further comprising a communication module to send said packet fragments to said second address.

13. (Currently Amended) A system to manage packet fragmentation for an address translation device, comprising:

a source node to send packet fragments for a packet having a first address, wherein each packet fragment includes a packet fragment header having a packet offset value and a more bit, said more bit set to a predetermined position indicating that more packet fragments are to follow; and

an intermediate node to receive said packet fragments and to translate said first address to a second address without reassembling said packet fragments into said packet;

wherein said intermediate node is further adapted to index offset values from each packet fragment in a verification table to determine whether all packet fragments for said packet have been received, each of said offset values representing a position from a starting position to an ending position of said packet, said intermediate node to determine a status of said more bit for each packet fragment, collect offset values in said verification table, index collected offset values by position in said verification table, and evaluate said collected offset values to identify any missing positions between said starting position and said ending position;

wherein said intermediate node is further adapted to send a verification message once all entries in said verification table are filled and to translate said first address into said second address, without reassembling said packet fragments into said packet, in response to said verification message.

14. (Original) The system of claim 13, further comprising a destination node having said second address to receive said packet fragments and reassemble said packet fragments into said packet.

15. (Currently Amended) A system to manage packet fragmentation for an address translation device, comprising:

a computer platform adapted to manage packet fragmentation;

said platform being further adapted to receive a plurality of packet fragments for a packet having a first address, translate the first address into a second address without reassembling said packet fragments into said packet, and send said packet fragments using said second address, wherein each packet fragment includes a packet fragment header having a packet offset value and a more bit, said more bit set to a predetermined position indicating that more packet fragments are to follow;

wherein said platform is further adapted to index offset values from each packet fragment in a verification table to determine whether all packet fragments for said packet have been received, each of said offset values representing a position from a starting position to an ending position of said packet, said intermediate node to determine a status of said more bit for each packet fragment, collect offset values in said verification table, index collected offset values by position in said verification table, and evaluate said collected offset values to identify any missing positions between said starting position and said ending position;

wherein said platform is further adapted to send a verification message once all entries in said verification table are filled and to translate said first address into said second address, without reassembling said packet fragments into said packet, in response to said verification message.

16. (Original) The system of claim 15, wherein said platform is further adapted to perform said translation by identifying a packet fragment having a packet header, with said packet header having a packet identifier, translation information and a packet length, determining whether all packet fragments for said packet have been received, retrieving

translation information from said packet header, and translating said first address into said second address using said translation information.

17. (Previously Presented) The system of claim 15, wherein said platform is further adapted to retrieve a packet length for said packet, and determine whether all positions for said packet are filled by said collected offset values using said packet length.

18. (Currently Amended) An article comprising:

a ~~storage~~ computer readable medium;

said ~~storage~~ computer readable medium including stored computer program instructions that, when executed by a ~~processor~~ computer, result in receiving a plurality of packet fragments for a packet having a first address, translating said first address into a second address without reassembling said packet fragments into said packet, and sending said packet fragments using said second address, wherein each packet fragment includes a packet fragment header having a packet offset value and a more bit, said more bit set to a predetermined position indicating that more packet fragments are to follow;

wherein the stored computer program instructions, when executed by a ~~processor~~ computer, further result in indexing offset values from each packet fragment in a verification table to determine whether all packet fragments for said packet have been received, each of said offset values representing a position from a starting position to an ending position of said packet, said intermediate node to determine a status of said more bit for each packet fragment, collect offset values in said verification table, index collected offset values by position in said verification table, and evaluate said collected offset values to identify any missing positions between said starting position and said ending position;

wherein the stored instructions, when executed by a computer, further result in sending a verification message once all entries in said verification table are filled and translating said first address into said second address, without reassembling said packet fragments into said packet, in response to said verification message.

19. (Currently Amended) The article of claim 18, wherein the stored computer program instructions, when executed by a ~~processor~~ computer, further result in said translating by identifying a packet fragment having a packet header, with said packet header having a packet identifier, translation information and a packet length, determining whether all packet fragments for said packet have been received, retrieving translation information from said packet header, and translating said first address into said second address using said translation information.

20. (Currently Amended) The article of claim 19, wherein the stored computer program instructions, when executed by a ~~processor~~ computer, further result in retrieving a packet length for said packet, and determining whether all positions for said packet are filled by said collected offset values using said packet length.

21. (Currently Amended) The article of claim 18, wherein the stored computer program instructions, when executed by a ~~processor~~ computer, further result in detecting an occurrence of a terminating condition prior to receiving all of said packet fragments for said packet, and releasing said packet fragments in accordance with said detection.